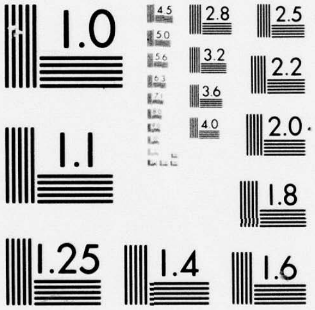
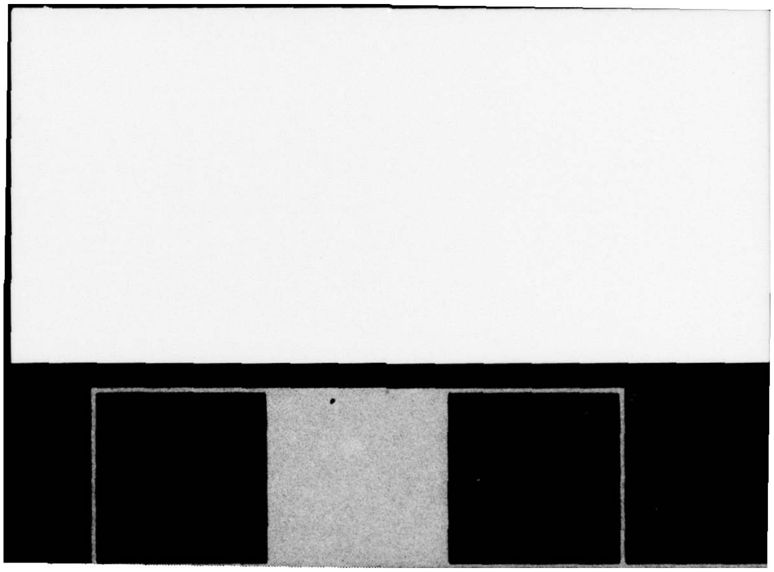
AD-A041 104 COLORADO UNIV BOULDER COMPUTING CENTER F/G 9/2
CSC PLOTTING PACKAGE.(U)
NOV 76 J R WARNER, K I JOY DAAG29-76-6-0095
UNCLASSIFIED UCCC-76-21 ARO-13742.1-EL NL

| OF |
AD
A041104

END
DATE
FILMED
8—77



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



ACCESSION for
NTIS White Section ☑
DDC Buff Section □
UNANNOUNCED □
JUSTIFICATION........................
....
BY..
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

10

ARO-13742.1-EL

CSC PLOTTING PACKAGE

J. R. Warner
K. I. Joy

November 1976

U. S. ARMY RESEARCH OFFICE

G
ARO Grant #DAAG29-76-0095

DDC
RECEIVED
JUN 27 1977
D

Computing Center
University of Colorado
Boulder, Colorado 80309

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

6 CSC PLOTTING PACKAGE.

~~5. TYPE OF REPORT & PERIOD COVERED~~

9 Final rept.

~~6. PERFORMING ORG. REPORT NUMBER~~

UCCC Report #76-21

7. AUTHOR(s)

10 J. R. Warner ~~and~~ K. I. Joy

8. CONTRACT OR GRANT NUMBER(s)

DAAG29-76-G-0095 new

9. PERFORMING ORGANIZATION NAME AND ADDRESS

~~The Regents of the~~
University of Colorado
Boulder, Colorado 80309

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

P-13742-A-EL

11. CONTROLLING OFFICE NAME AND ADDRESS

U. S. Army Research Office
Post Office Box 12211
Research Triangle Park, N. C. 27709

~~12. REPORT DATE~~

11 November 1976

13. NUMBER OF PAGES

53

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

Robert P. Thacker
ONR Representative
6740 E. Hampden Avenue
Denver, Colorado 80222

12 56p.

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

14 UCCC-76-21

Approved for public release; distrubution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

15 ✓ DAAG29-76-G-0095

18 ARO

18. SUPPLEMENTARY NOTES

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Graphics compatibility capabilities

19 13742.1-EL

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The CSC plotting package is a computer software system for generating tabular graphs from United States Army personnel information. As input, the CSC plotting package accepts a sequential file of information built using the United States Army's SIR (Selective Information Retrieval) system. As output, the package generates an annotated histogram or pie chart derived from the sequential SIR file.

→ next page

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

410260

Block 20

cont

SIR allows users to selectively define and build a retrieval file from an information data bank. The CSC plotting package synthesizes the retrieval file in generating a customized plot.

The CSC plotting package accepts simple input commands for generating the tabular plots. Graphics output may be routed to any display device that supports line drawing capabilities. The plot package is functional in both batch and time-sharing environments.

The CSC plotting package is written in ANSI Standard FORTRAN IV for compatibility among computing installations.

The following sections of this report detail the general configuration of the package. Four appendices:

1) describe the user interface (i.e., input commands);
2) detail a sequence of example plots;
3) describe the device-dependent graphics routines; and
4) provide the general system schematic.

CSC PLOTTING PACKAGE

by

J. R. Warner
K. I. Joy

UCCC Report #76-21

Final Report for ARO Grant
#DAAG29-76-G-0095 entitled
"Graphics Compatibility System (GCS)"

Computing Center
University of Colorado
Boulder, Colorado 80309

November 1976

D D C
JUN 27 1977

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

1. Introduction

The CSC plotting package is a computer software system for generating tabular graphs from United States Army personnel information. As input, the CSC plotting package accepts a sequential file of information built using the United States Army's SIR (Selective Information Retrieval) system. As output the package generates an annotated histogram or pie chart derived from the sequential SIR file.

SIR allows users to selectively define and build a retrieval file from an information data bank. The CSC plotting package synthesizes the retrieval file in generating a customized plot.

The CSC plotting package accepts simple input commands for generating the tabular plots. Graphic output may be routed to any display device that supports line drawing capabilities. The plot package is functional in both batch and time-sharing environments.

The CSC plotting package is written in ANSI Standard FORTRAN IV for compatibility among computing installations.

The following sections of this report detail the general configuration of the package. Four appendices:

1) describe the user interface (i.e., input commands);

2) detail a sequence of example plots;

3) describe the device-dependent graphics routines;

4) provide the general system schematic.

2. Overview of the Plotting System

The plotting system is composed of four distinct program modules. Each module is composed of one or more FORTRAN subroutines with a small set of machine language primitives. Inter-program communication is via labeled COMMON blocks. This structuring simplifies program overlaying that may be required on smaller computing systems.

Figure 2-1 identifies the modular system components and the communication links among them.

The Monitor sequentially invokes each of the modules of the plotting system. It is a compact program whose only function is to invoke the modular tasks. In an overlaid implementation, the monitor would correspond to the primary overlay.

The Sequential Data File contains four header records that fully define the content and format of the remaining data. A full description of this file with the header records is given in Section 3.

The Data File Definition Module reads the sequential data file header records, storing the required file information in global COMMON blocks.

The user input commands are simple directives to the plotting system that define the explicit plot to be generated. A classification of these commands with their syntax is given in Section 4. The Command Parsing Module interprets the user's plot generation directives, checking for compatibility with the data file content. Syntactical errors and data file incompatibilities are flagged as errors.

A valid set of commands completely defines a unique plot. The command information is stored in global COMMON blocks.

The Data Synthesis Module utilizes the data file header record information coupled with the user's plot directives establishing the graph plotting information. The actual data records of the data file are sequentially processed. The graphing criteria provided by the user is applied to the data resulting in tabular plotting information. This plot data, along with titling and other graph enhancements, is stored in the plotting information COMMON blocks. A detailed discussion of this data synthesis process is given in Section 5.

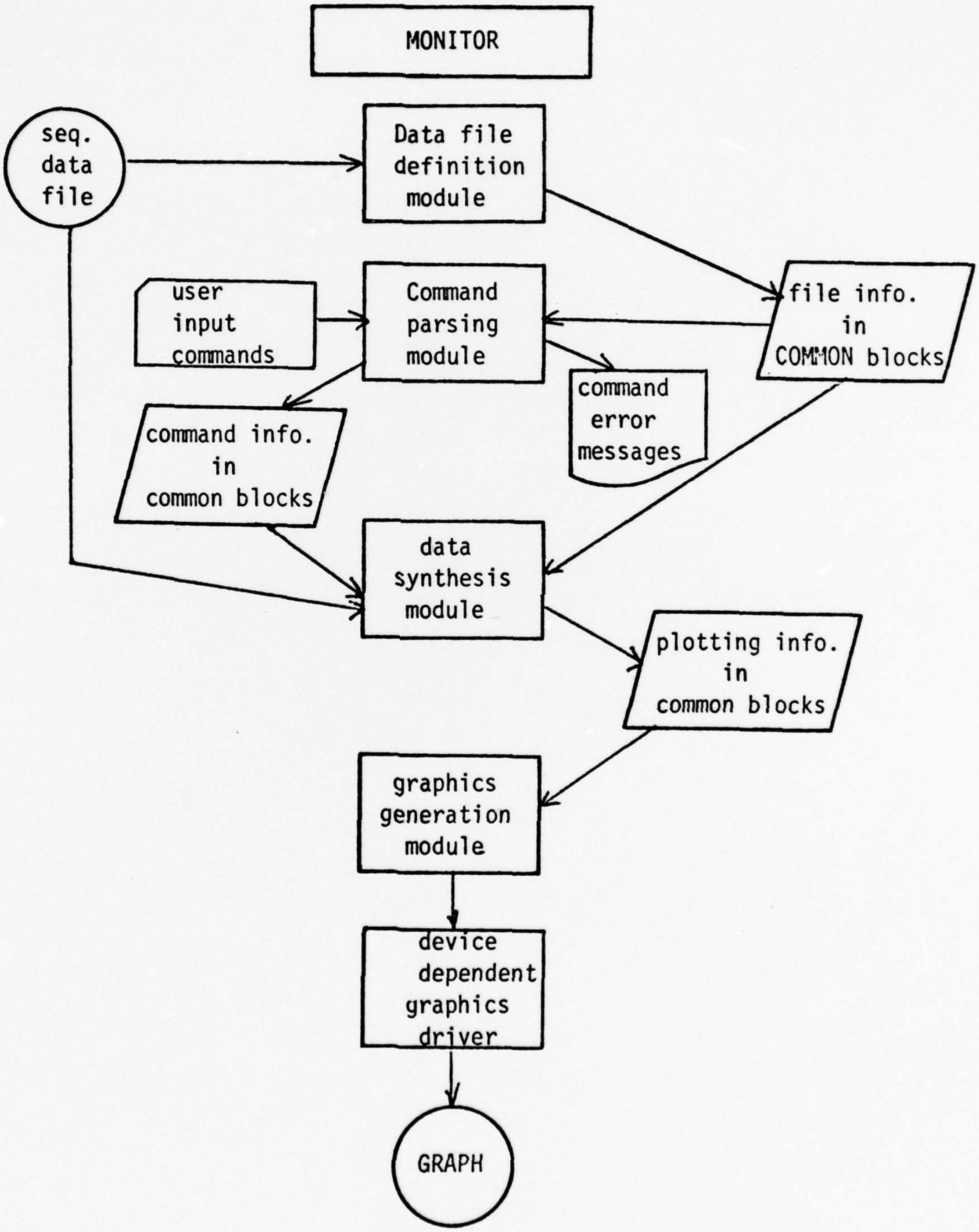

MONITOR
seq. data file
Data file definition module
user input commands
Command parsing module
file info. in COMMON blocks
command error messages
command info. in common blocks
data synthesis module
plotting info. in common blocks
graphics generation module
device dependent graphics driver
GRAPH

Figure 2-1

CSC PLOTTING SYSTEM FLOW CHART

The Graphics Generation Module simply interprets the COMMON plotting information to generate the final display graph (either histogram or pie chart). The Device Dependent Graphics Driver is the low-level set of routines for actually plotting on the active graphics device. Details of this module, including certain efficiency enhancements, are discussed in Appendix C.

3. The Data File Definition Module

The sequential data file is the hub of the plotting system. This file maintains textual and quantitative information on one or more descriptors. A descriptor is a category of information, for example, NAME, SOCIAL SECURITY NUMBER, BIRTH DATE, SALARY, etc. The number of descriptors in the data file is a function of the SIR query used to generate the data file. Descriptors are further categorized by the values they may have. For example, the descriptor, RANK, may have values SGT, MAJ, CPT, LTC, etc. These values are the states that a descriptor may have, or the descriptor states.

Depending on the type of information in the descriptor, the number of possible descriptor states may vary. For example, the descriptor NAME will probably have as many descriptor states as there are records in the file, whereas RANK will have multiple occurrences of a small number of states. Numeric descriptors, such as SALARY, or BIRTH DATE, will rarely have more than one occurrence of the same state. However, it is often useful to reference groups of quantitative data. For example, SALARY: from $800.00/month to $1000.00/month; or BIRTH DATE: from 47/06/03 to 50/06/03. This grouping of quantitative data is often useful in generating tabular graphs.

The header records of the data file define the content and format of the data file descriptors. This includes the name of the descriptor, its type (alphanumeric or quantitative), the maximum number of states in the descriptor, and other information defining the descriptor state format within each data record.

The Data File Definition Module interprets the header records, saving the descriptor names and other relevant data file information in global COMMON blocks. These COMMON blocks are used by subsequent plotting modules for command validity checking and to monitor the reading of the data file.

4. The Command Parsing Module

Two general graphic display types are available to the user: the histogram and the pie chart. Each type of graph involves tabulating occurrences of states within a descriptor. These occurrences may be single-valued as in alphanumeric descriptors, or they may define a range of values as in quantitative descriptors.

To generate a plot, the user must define both the tabulation descriptors and the states to be tabulated. Certain graph enhancements are often desirable, such as primary titling, auxiliary titling, tabulation counts, tabulation percentages, etc. Facility is also made for tabulations on one descriptor as a function of the states of another qualifying descriptor. For example, tabulating on rank for only those personnel born between 1948 and 1952.

The Command Parsing Module accepts free format command lines for defining the tabulation descriptors, tabulation states, and any plot qualifiers or enhancements. Command lines consist of a command keyword and possibly additional keyword parameters. The keyword is separated from the parameters by a colon. Parameters are separated by commas. The end of parameters is denoted by an asterisk. Some commands may require more than one 80-column input line.

Facility is made for interactive plot generation on those machines with interactive capabilities.

Commands are parsed and interpreted one line at a time. Syntactical checking flags obvious errors. All descriptor and state references are verified against the data file COMMON block.

Output from the command parsing module is COMMON blocks that define the tabulation criteria and plot layout. This information acts as input to the data synthesis module.

The entire system, with the exception of a few machine language primitives, is written in ANSI FORTRAN The code has been verified using the PFORT verifier. Non-standard statements are flagged in the source code.

Portability is somewhat a mixed blessing when designing and implementing large-scale production computer programs. It is highly desirable if many machine configurations are to execute the system. However, because machine language functions and non-standard FORTRAN constructs may not be used, execution efficiency is often significantly eroded.

Certain key routines in the plotting system, especially those related to parsing and I/O, would be better implemented using local machine-dependent capabilities. These routines are marked in the source code.

5. The Data Synthesis Module

The information stored in COMMON by the Command Parsing Module is utilized to sequentially read and process the data file and to generate tabular information for the Graphics Generation Module. Each record of the data file is examined according to the descriptor information supplied by the user.

If a qualifying descriptor was specified, the qualifying descriptor state (which appears in each record) is compared against the allowable qualifying states. If the state is found to be among these allowable states, the plot descriptor state is then processed.

The plot descriptor state is compared against the allowable plot states. If a suitable comparison is made, tabulation counts are incremented for this descriptor state.

An error message is output if, upon processing the entire file, no tabulations occurred (or all tabulations were in the "other" category). The error message will contain the basic information that was passed to the Data Synthesis Module via the Labeled Common.

The Data Synthesis Module will only be invoked if tabulations are to be generated or regenerated. If the command stream contains no plot type command (i.e., HISTO, HISTOV, PIE CHART, or STATES), the Graphics Generation Module is immediately called. This enables the user to generate multiple plots without having to reread the data file. See Appendix B for detailed examples.

6. The Graphics Generation Module

The tabular information passed by the Data Synthesis Module is used along with user specified parameters to generate the final plot.

The plot may take one of three basic forms--pie chart, histogram with horizontal bars, or histogram with vertical bars. The user specified titles and tabular state displays are generated with software text fonts that have a variable text size. The Device Dependent Graphics Driver is invoked to display the final plot on the active display device.

The basic form of the pie chart is displayed in Figure 6-1.

The basic form of the "histogram with horizontal bars" is displayed in Figure 6-2.

The basic form of the "Histogram with Vertical bars" is given in Figure 6-3.

These basic shapes may be reduced in size to accommodate titles, quantities, percents, or unusual length in the tabular data.

The Graphics Generation Module will generate the basic pie chart or histogram along with the titles, quantities or percents, and all alphanumeric tabular data pertaining to each output state. (See the examples in Appendix B.)

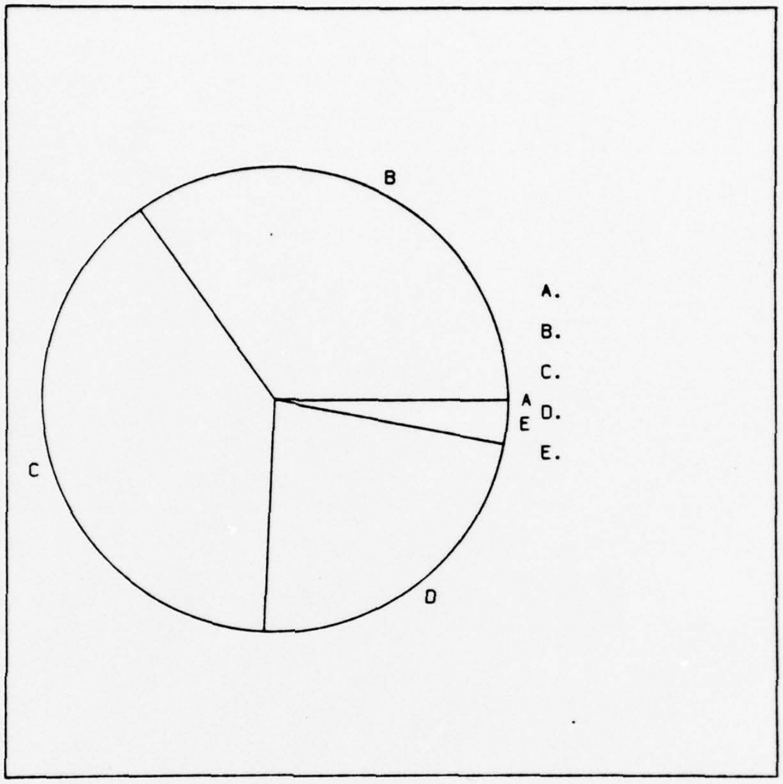


Figure 6-1

PIE CHART

The tabulated states are represented as wedges of the pie. Wedges are identified alphabetically. A key to the right of the pie defines the actual state names (or ranges) associated with each wedge. The Main Title is centered at the top of the display; the X-Title is centered below the pie.

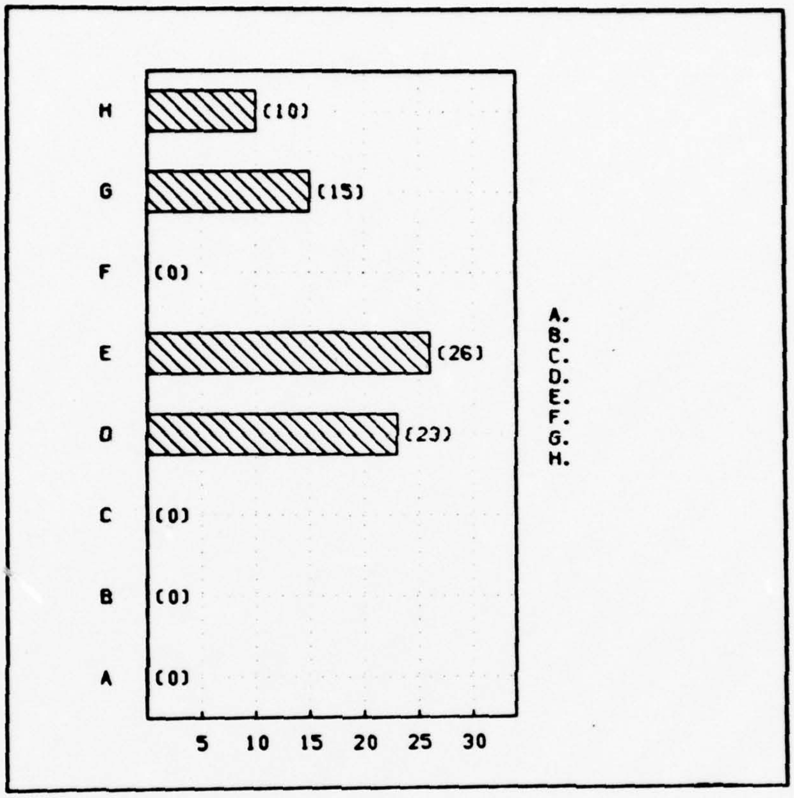


Figure 6-2

HISTOGRAM WITH HORIZONTAL BARS

The tabulated states are represented as shaded bars identified alphabetically from the bottom of the display. A key to the right of the chart defines the actual state names (or ranges) associated with each bar. This type of chart allows up to 29 horizontal bars equally spaced along the left border of the display.

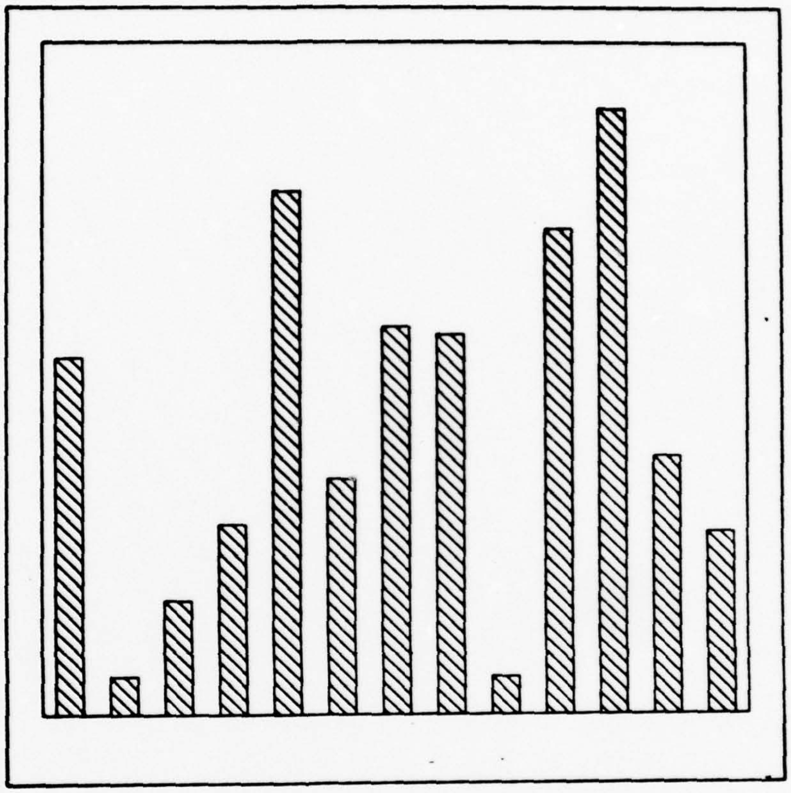


Figure 6-3

HISTOGRAM WITH VERTICAL BARS

The tabulated states are represented as shaded bars extending from the bottom of the display. The state names are centered below each bar if the printing will remain legible. If the names would have to be reduced in size below a legibility minimum, an alphabetic identifier is assigned to each vertical bar. These identifiers are then defined in a key to the right of the plot, similar to the horizontal histogram key. The chart may contain up to 29 equally spaced vertical bars.

7. Conclusions

The CSC Plotting System is a versatile system for generating concise tabular graphs. It is designed as a modular extension to the SIR information retrieval system. The file definition capabilities are easily amended to process any sequential file of descriptor/state data.

The graph layout and annotation facilities are maintained wholly in software allowing output to any graphics device with vector-drawing capabilities.

APPENDIX A - SUMMARY OF COMMANDS

A.1 Command Classes

Plotting System commands may be separated into three categories. Each category identifies a distinct class of required or optional commands.

1) Required Commands: Certain commands are required by the plotting system to produce a tabular graph. The user must define the type of graph (histogram or pie chart) and the dependent descriptor by using one of the following commands: HISTO, HISTOV, PIE CHART. Next, the user must explicitly give the descriptor states or descriptor state ranges to be associated with the dependent descriptor. This is done via the STATES command. The PLOT command must be given to generate the tabular graph. Finally, the END command must be the last command in the input stream. END indicates the end of plotting system commands.

   The minimum command stream to generate a plot is:

```
HISTO:   [HISTOV]   [PIE CHART] *
STATES:
PLOT:
END:
```

2.) Plot Enhancement Commands: A series of optional commands is provided to improve graph readability. Annotation capabilities include axes labeling, primary titles, and the date the plot is generated. Facility is also made for adding tabulation counts and percentages, and the generation of grid lines on histograms.

The plot type commands (HISTO, HISTOV, and PIE CHART) have optional parameters that allow qualification on the acceptable states of the plot descriptor. The general format for descriptor qualifications is:

Type Keyword: primary descriptor, WITH, qualifying descriptor, qualifying descriptor state(s)*, (multiple states are separated by commas).

Examples are given under the HISTO command.

3) Informational Commands: The FILE DATA command provides information on the contents of the sequential data file. It is especially useful interactively, giving the descriptor names, types (alphanumeric or quantitative), and the maximum number of descriptor states.

A.2 Command Syntax

Each command to the Plotting System consists of one or more free format input lines. In batch mode these input lines are punched on cards. Interactively, an input line is simply one typed line of information.

All commands start with a command directive followed by a colon : , e.g. PIE CHART:. Directives must be spelled correctly, but spacing between successive words of multiple word directives is arbitrary.

Most commands require one or more parameters to further define the command directive. Parameters are entered in free format following the colon. Multiple parameters must be separated by commas. The end of parameters is denoted by an asterisk * . All commands that require parameters must be terminated by an asterisk.

Four commands, HISTO:, HISTOV:, PIE CHART:, STATES:, may be continued on successive input lines. However, parameters may not overlap between successive input lines.

A.3 Required Commands

There are six required commands. Three of these, HISTO, HISTOV, and PIE CHART are plot type commands. The remaining three commands, STATES, PLOT, and END are also required.

The user must give one of three plot type commands indicating the type of tabular graph to be generated. Each command has the general format:

Plot Type Command: Primary Descriptor [, WITH,
Qualifying Descriptor, Qualifying Descriptor State(s)] *

An example of qualifying descriptor usage is given in the description of HISTO, below.

The primary descriptor may either be spelled out exactly or abbreviated. Abbreviations correspond to the ordinal position of the descriptor within the data file records. Abbreviations are of the form, .*n*., where *n* is the ordinal position of the descriptor in the record. For example, if each record contained, in order, the descriptors:

NAME
SSN
BIRTH DATE
SALARY
LOSS DATE
GRADE

these descriptors may be abbreviated:

Descriptor Name	Abbreviation
NAME	.D1.
SSN	.D2.
BIRTH DATE	.D3.
SALARY	.D4.
LOSS DATE	.D5.
GRADE	.D6.

The ordering of descriptors within the data file and the corresponding descriptor abbreviations are available to the user via the FILE DATA command (Section A.5).

The Plot Type Commands may be continued on successive input lines. Continuation lines are most useful when a qualifying descriptor is used to delimit the primary descriptor tabulations. For readability, a qualified plot type command might have the format:

```
Plot Type Command:  Primary Descriptor, WITH,
  Qualifying Descriptor,
    Qualifying Descriptor State-1,
    Qualifying Descriptor State-2,
                 .
                 .
                 .
    Qualifying Descriptor State-n*
```

The keyword, "WITH", must be included in the command line if a qualifying descriptor is to be used.

All examples under the plot type commands will be generated from the following sample data file:

Descriptor Name	Abbreviation	Type	Maximum # of States
NAME	.D1.	alpha	9999
AGE	.D2.	quantitative	99
GRADE	.D3.	alpha	12
BIRTH DATE	.D4.	quantitative	999999
LOSS DATE	.D5.	quantitative	999999
HEIGHT (meters)	.D6.	quantitative	9999
WEIGHT (meters)	.D7.	quantitative	9999

Note that GRADE is the only descriptor with a "manageable" number of states. "Manageable" implies that all states may be included on a single plot.

A.3.1 HISTO

This plot type command defines the next generated graph to be a horizontal histogram. Horizontal implies that the bars of the histogram will extend from left to right beginning at the left border of the graph.

Command Format:

```
HISTO:  Primary descriptor [, WITH,
     Qualifying descriptor,
        Q - descriptor - state - 1,
        Q - descriptor - state - 2,
                   .
                   .
                   .
        Q - descriptor - state - n] *
```

Examples:

```
1)  HISTO:  GRADE *
```

---the equivalent abbreviated command is ---

```
2)  HISTO:  .D3. *
```

-either command will generate a horizontal histogram on the primary descriptor, GRADE.

```
3)  HISTO:  GRADE, WITH, LOSS DATE,
            760000, 769999 *
```

-this command generates a histogram on the dependent descriptor, GRADE. However, the histogram will only tabulate on individual records where the loss date is during calendar year 1976. All other records will be disregarded in the tabulation.

```
4)  HISTO:  LOSS DATE, WITH, GRADE,
                        LT1,
                        LT2,
                        CPT,
                        MAJ,
                        LTC *
```

-this command generates a histogram on the quantitative descriptor, LOSS DATE. Only those records with GRADE, LT1, LT2, CPT, MAJ, and LTC are tabulated.

* * * * * NOTE * * * * *

In all the above examples, only the dependent descriptor name has been declared. The user must still use the STATES: command to define the explicit states or state ranges that are to be tabulated for the dependent descriptor.

A.3.2 *HISTOV*

This plot type command defines the next generated graph to be a vertical histogram. Vertical implies that the bars of the histogram will extend from bottom to top beginning at the bottom border of the graph.

```
Command Format:  HISTOV:  Primary descriptor  [, WITH,
                       Qualifying descriptor,
                          Q - descriptor - state - 1,
                          Q - descriptor - state - 2,
                                    .
                                    .
                                    .
                          Q - descriptor - state - n] *
```

Examples: (See HISTO: A.3.1)

A.3.3 PIE CHART

This plot type command defines the next generated graph to be pie chart.

Command Format:
```
PIE CHART: Primary descriptor [, WITH,
      Qualifying descriptor,
         Q - descriptor - state - 1,
         Q - descriptor - state - 2,
                     .
                     .
                     .
         Q - descriptor - state - n] *
```

Examples: (See HISTO: A.3.1)

A.3.4 STATES

This required command defines the explicit states or state ranges of the primary descriptor that are to be tabulated. Explicit state values for alphanumeric type descriptors must be spelled out completely. State ranges, used to delimit quantitative or numeric type descriptors are defined FROM an initial value TO a final value.

Two keywords may be used in defining descriptor states:

.ALL. --- will perform tabulations on all explicit states of alphanumeric-type descriptors. .ALL. may not be used with quantitative or date-type descriptors. It is only useful when the dependent descriptor has a small number of possible states (e.g. GRADE from the sample data file).

.OTHER.-- will tabulate on all states or state ranges not explicitly defined in the STATES command. This keyword may be used with either alphanumeric or quantitative descriptors.

Command Format:

(alphanumeric primary descriptor)

```
    STATES:  state - 1, state - 2, state - 3, ..., state - n*
```

(quantitative primary descriptor)

```
    STATES:  from - value - 1, to - value - 1,
             from - value - 2, to - value - 2,
                                  .
                                  .
                                  .
             from - value - n, to - value - n  *
```

In either format the explicit states or state ranges may be continued on successive input lines.

Examples:

1) (with primary descriptor, GRADE)

STATES: LT1, LT2, CPT, MAJ, LTC *

---defines five explicit tabulation states for descriptor, GRADE. Only those records with GRADE equal to one of these five values will be tabulated. The corresponding graph would have five rows (HISTO), five columns (HISTOV), or five wedges (PIE CHART).

2) STATES: LT1, LT2, CPT, MAJ, LTC, .OTHER. *

---defines five explicit tabulation states and a sixth state for all other state values. All records in the data file are included in the tabulation. All records not having one of the explicit state values are counted in the .OTHER. category. The output plot will have six tabulation components (rows, columns, or wedges).

3) STATES: .ALL. *

---will tabulate on each explicit state of the primary descriptor. The number of rows/columns/wedges will equal the maximum number of states in the descriptor. (See the FILE DATA command, A.5, for determining the maximum number of descriptor states.)

4) (with primary descriptor HEIGHT)

```
STATES: 1.5, 1.6,
        1.6, 1.7,
        1.7, 1.8,
        1.8, 1.9,
        1.9, 2.0 *
```

---will generate five tabulation rows/columns/wedges based on the five height ranges. Records with height less than 1.5 meters or greater than 2.0 meters will not be tabulated.

5) STATES: 1.5, 1.6,
           1.6, 1.7,
           1.7, 1.8,
           1.8, 1.9,
           1.9, 2.0,
           .OTHER. *

---will generate six tabulation components. All records in the data file will be tabulated. All records with height less than 1.5 meters or greater than 2.0 meters will be tabulated in the .OTHER. component.

6) (with primary descriptor LOSS DATE)

STATES: 750000, 769999,
        770000, 789999,
        790000, 809999,
        810000, 999999 *

---will generate four tabulation components:

component - 1: 1975 - 1976
component - 2: 1977 - 1978
component - 3: 1979 - 1980
component - 4: 1981 - 1999

Records with LOSS DATE prior to 1975 will not be tabulated.

A.3.5 PLOT

This command indicates the end of plot setup commands. The data file is processed, tabulating on the specified states or state ranges of the primary descriptor. The actual plot is then generated on the active plot medium.

Command Format: PLOT: (no parameters)

A.3.6 END

This command indicates the end of plotting system commands. It terminates the plotting system program. END should be the last command to the plotting system.

Command Format: END: (no parameters)

A.4 Plot Enhancement Commands

There are eight plot enhancement commands, MAIN TITLE, XTITLE, YTITLE, DATE, COUNT, QUANT, PERCT, GRID.

A.4.1 MAIN TITLE

This command defines a one-line title for the graph. The title appears centered at the top of the graph. The title may be up to 30 characters long.

Command Format: MAIN TITLE: title *

Example: MAIN TITLE: STRENGTH SUMMARY *

A.4.2 XTITLE

This command defines a one-line title to appear at the bottom of the plot. It will be centered below the X-axis of horizontal or vertical histograms, and directly below a pie chart. The title may be up to 30 characters long.

On horizontal histograms the X-axis will correspond to the tabulation counts. On vertical histograms the X-axis contains the descriptor states or state ranges.

Command Format: XTITLE: title *

Examples:

```
1) HISTO:  GRADE  *
   XTITLE:  TABULATION COUNTS  *

2) HISTOV: GRADE  *
   XTITLE:  PERSONNEL RANK  *
```

A.4.3 YTITLE

This command defines a one-line title oriented 90 degrees from the horizontal and centered to the left of the Y-axis of histograms. YTITLE will not appear in pie charts. The title may be up to 30 characters long.

On horizontal histograms the Y-axis will correspond to the descriptor states or state ranges. On vertical histograms the Y-axis contains the tabulation counts.

Command Format: YTITLE: title *

Examples:

```
1) HISTO:  GRADE *
   XTITLE:  TABULATION COUNTS  *
   YTITLE:  PERSONNEL RANK  *

2) HISTOV:  GRADE  *
   XTITLE:  PERSONNEL RANK  *
   YTITLE:  TABULATION COUNTS  *
```

A.4.4 DATE

This command allows the user to define an explicit date to appear in the lower right corner of the plot. If this command is omitted, the plotting system obtains the current computer system date, in the form YY/MM/DD, and prints it in the lower right corner.

Command Format: DATE: date string *

Examples:

1) DATE: 31 AUGUST 1977 *

2) DATE: OCTOBER 18, 1929 *

3) DATE: SATURDAY, JULY 10 1976 *

4) DATE: 31 JAN 74 *

A.4.5 COUNT

This command defines the interval on the tabulation axis of histograms at which tick marks are to be placed. The incremental quantity associated with each tick mark is written adjacent to the actual tick mark.

If this command is not given, no tick marks or incremental quantities are written on the tabulation axis.

Command Format: COUNT: *n* *
-where *n* is the tick mark interval.

Examples:

1) COUNT: 10 *
--will place tick marks and incremental quantities every ten units along the tabulation axis.

2) COUNT: 150 *
--will place tick marks and incremental quantities every 150 units along the tabulation axis.

3) COUNT: 0 *
--will suppress tick marks in all subsequent plots.

A.4.6 QUANT

This command will place the computed tabulation quantities next to each bar of a histogram or wedge of a pie chart. If this command is not given, no quantities are printed on the graph.

QUANT will generate quantity counts on all subsequent plots. To suppress quantity counts the NO QUANT: command should be used.

```
Command Format:        QUANT:  (no parameters)
                    NO QUANT:  (no parameters)
```

A.4.7 PERCT

This command calculates the tabulation percentage of each bar in a histogram or wedge of a pie chart. The percentage is printed on the plot adjacent to the bar or wedge.

PERCT generates tabulation percentages on all subsequent plots. To suppress tabulation percentages the NO PERCT: command should be used.

```
Command Format:       PERCT:  (no parameters)
                   NO PERCT:  (no parameters)
```

A.4.8 GRID

This command superimposes a checkerboard grid over a histogram. One set of grid lines is aligned with the descriptor state bars; the other set is aligned with the tabulation intervals. The grid often improves the readability of histograms having many tabulated states.

The grid will be superimposed over all subsequent histograms. To suppress grid generation, the NO GRID: command should be used.

Command Format:

```
GRID:
NO GRID:
```

A.5 Informational Commands

There is one informational command, FILE DATA.

A.5.1 FILE DATA

This command lists all relevant descriptor information, obtained from the data file header records, to the active printer device (e.g. the line printer in batch mode or the alphanumeric terminal in interactive mode).

FILE DATA prints out the following information on each descriptor:

1) The full descriptor name.

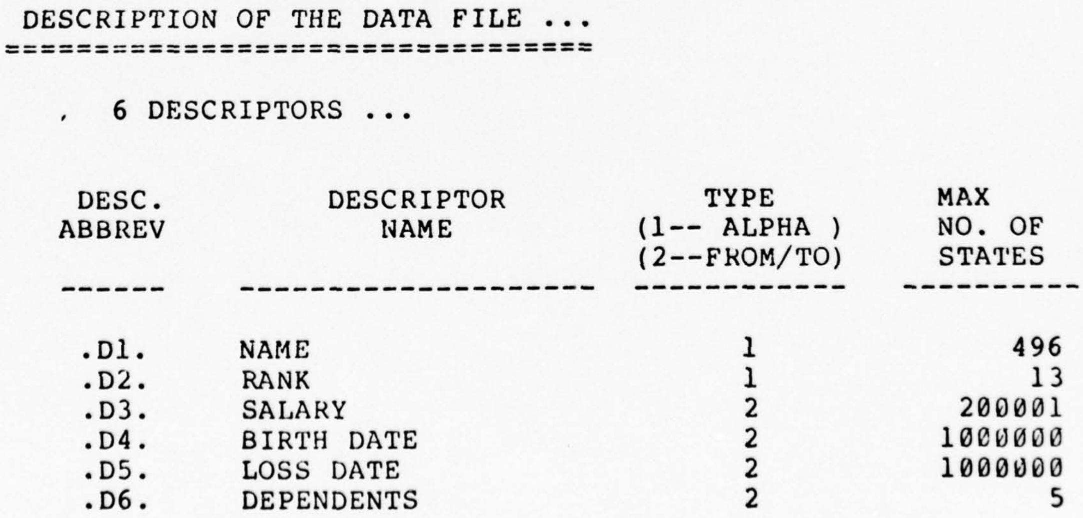
2) The abbreviation for the descriptor.

3) The type of the descriptor (alphanumeric or quantitative).

4) The maximum number of states for the descriptor.

The information is printed in a 72-column table.

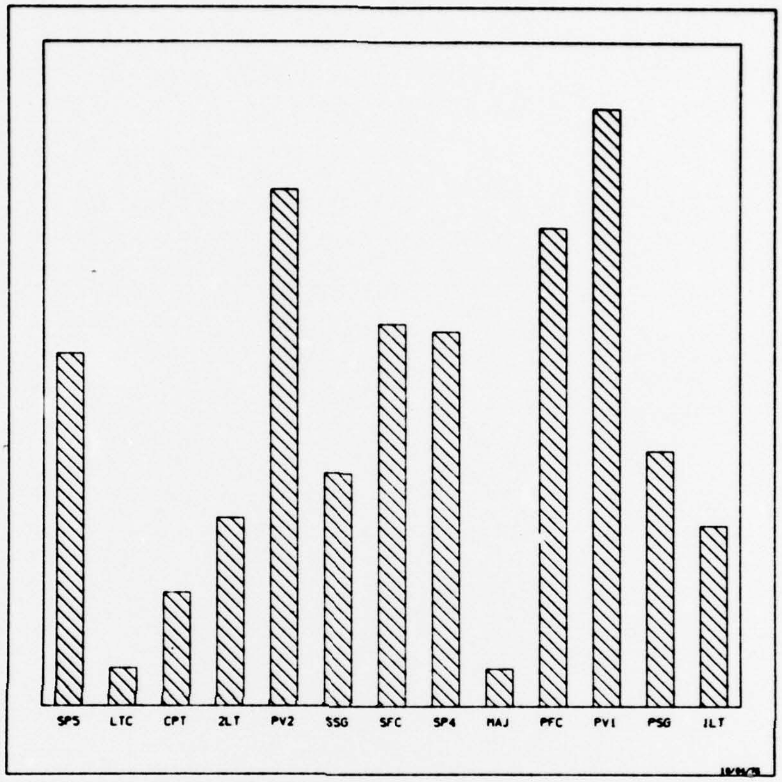
Command Format: FILE DATA: (no parameters)



```
DESCRIPTION OF THE DATA FILE ...
=================================

 6 DESCRIPTORS ...

  DESC.        DESCRIPTOR             TYPE         MAX
 ABBREV           NAME            (1-- ALPHA )   NO. OF
                                  (2--FROM/TO)   STATES
 ------   --------------------   ------------  ----------

  .D1.     NAME                        1              496
  .D2.     RANK                        1               13
  .D3.     SALARY                      2           200001
  .D4.     BIRTH DATE                  2          1000000
  .D5.     LOSS DATE                   2          1000000
  .D6.     DEPENDENTS                  2                5
```

APPENDIX B - CSCPLOT EXAMPLES

This section presents 8 example plots generated from the data file described on page A-23. These displays were generated on an FR-80 microfilm recorder and photographically enlarged for this publication.

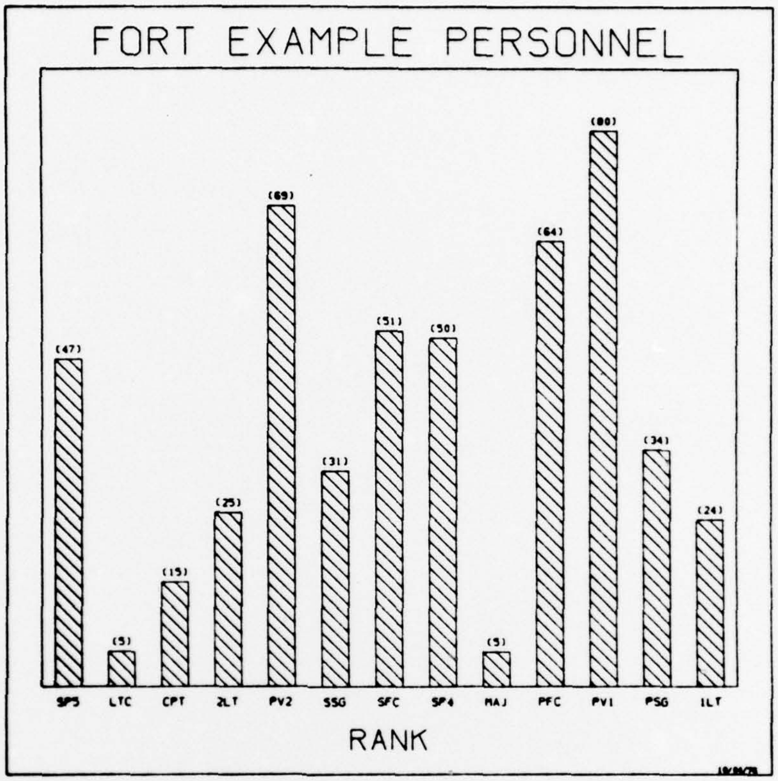


EXAMPLE 1

HISTØV: RANK *

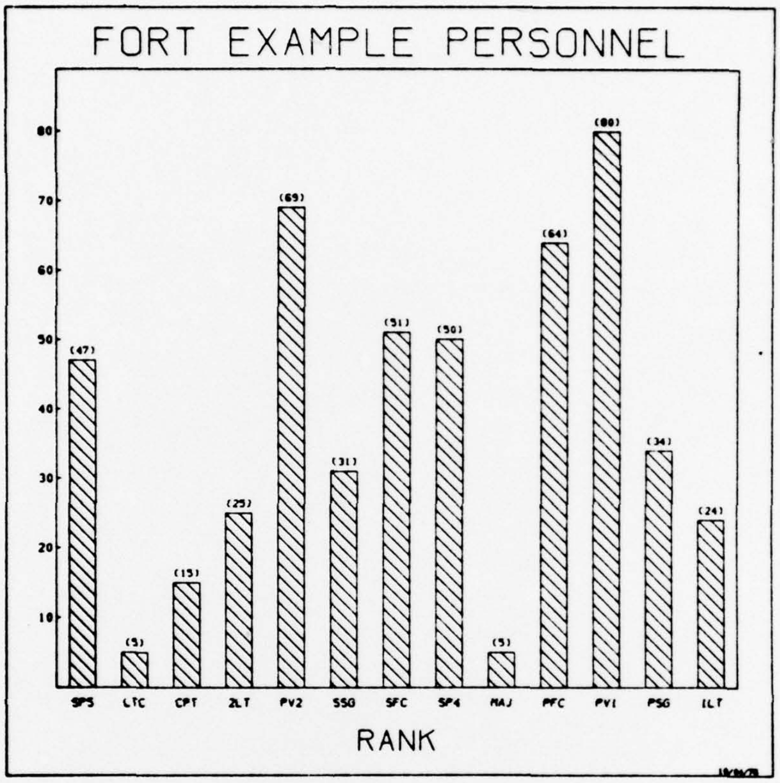
STATES: .ALL. *

PLØT:



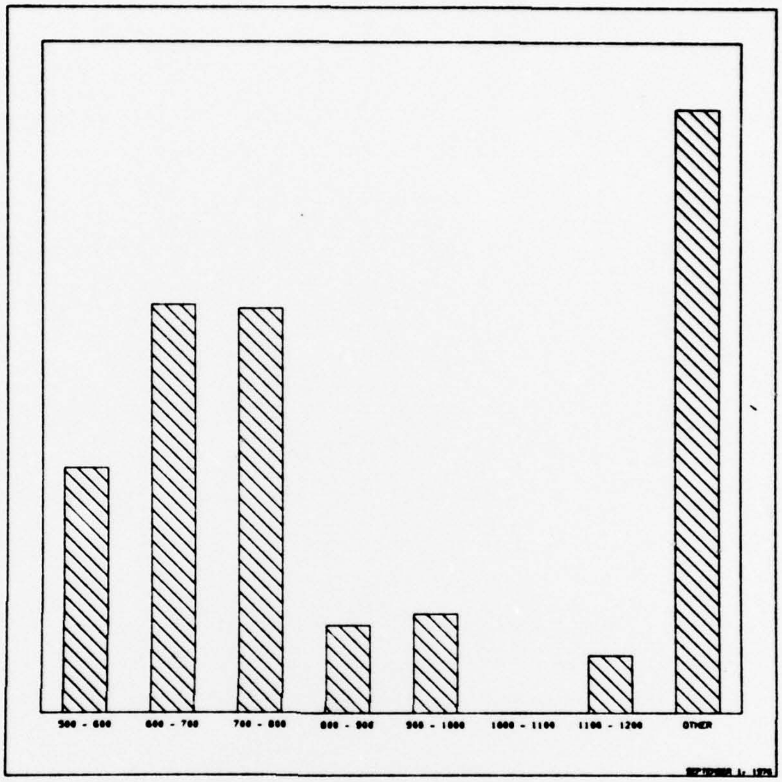
EXAMPLE 2

```
HISTØV:  RANK *
STATES:  .ALL.*
QUANT:
MAIN TITLE:  FORT EXAMPLE PERSONNEL *
XTITLE:  RANK *
PLØT:
```



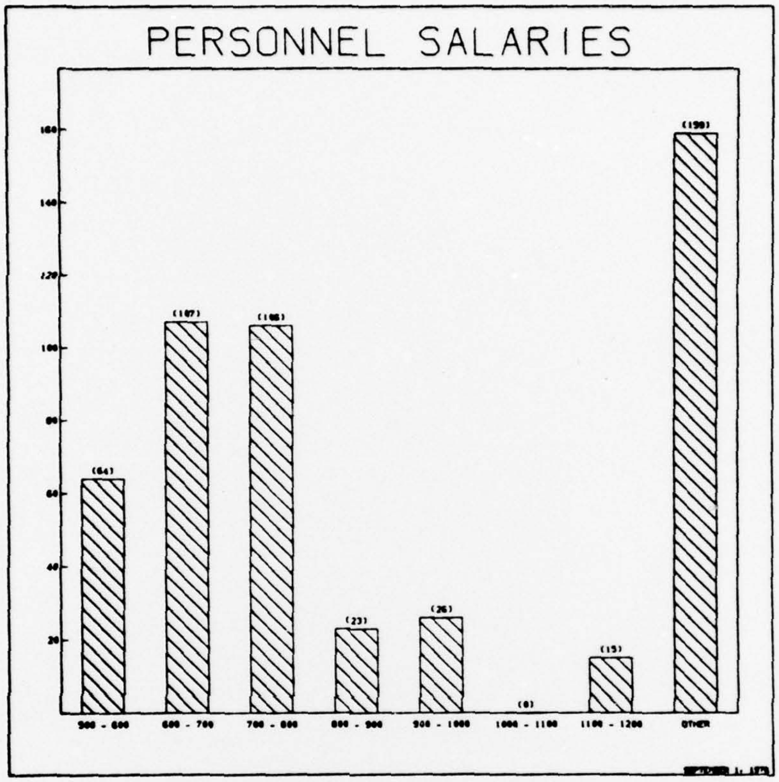
EXAMPLE 3

```
HISTØV:  RANK *
STATES:  .ALL *
QUANT:
MAIN TITLE:  FORT EXAMPLE PERSONNEL *
XTITLE:  RANK *
CØUNT:  10 *
GRID:
PLØT:
```



EXAMPLE 4

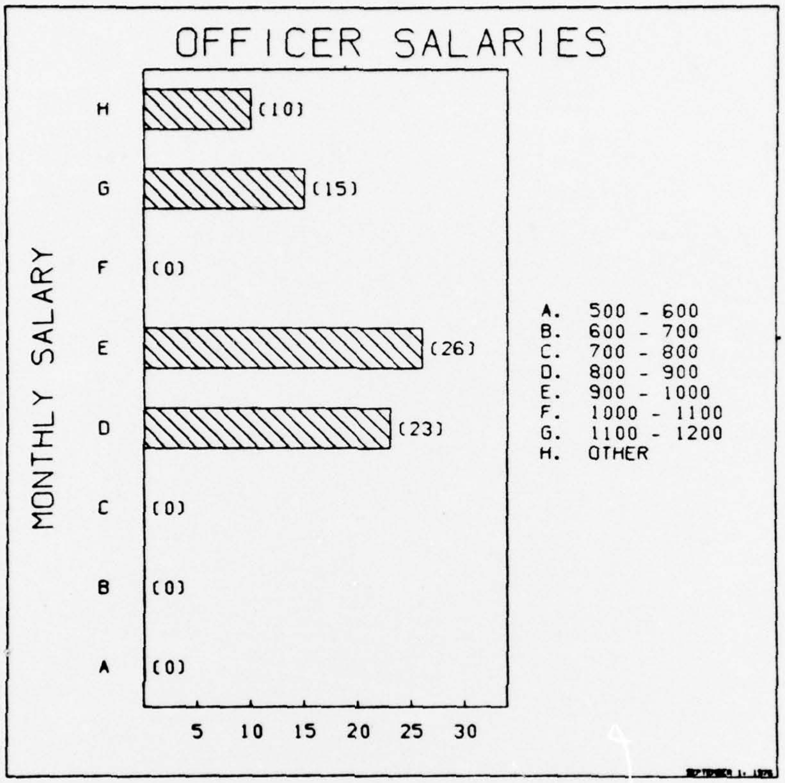
```
HISTØV:  SALARY *
STATES:  500, 600, 600, 700,
         700, 800, 800, 900,
         900, 1000, 1000, 1100,
         1100, 1200,
         .OTHER. *
DATE:  SEPTEMBER 1, 1976 *
PLØT:
```



EXAMPLE 5

(These commands are assumed to directly follow the commands of Example 4)

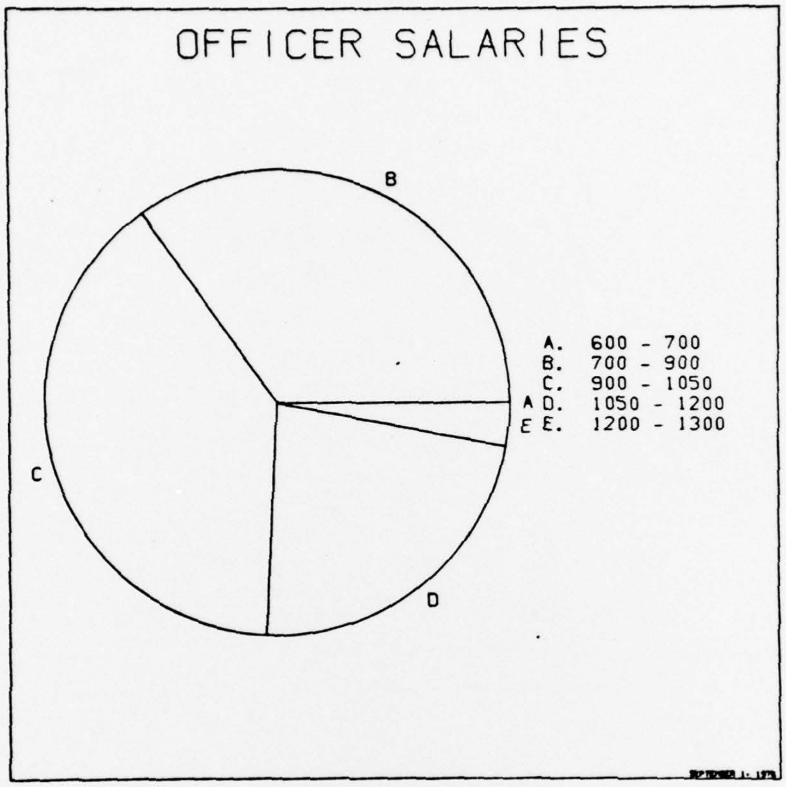
```
        .
        .
        .
        .
CØUNT:  20 *
GRID:
QUANT:
MAIN TITLE:  PERSONNEL SALARIES *
PLØT:
```



EXAMPLE 6

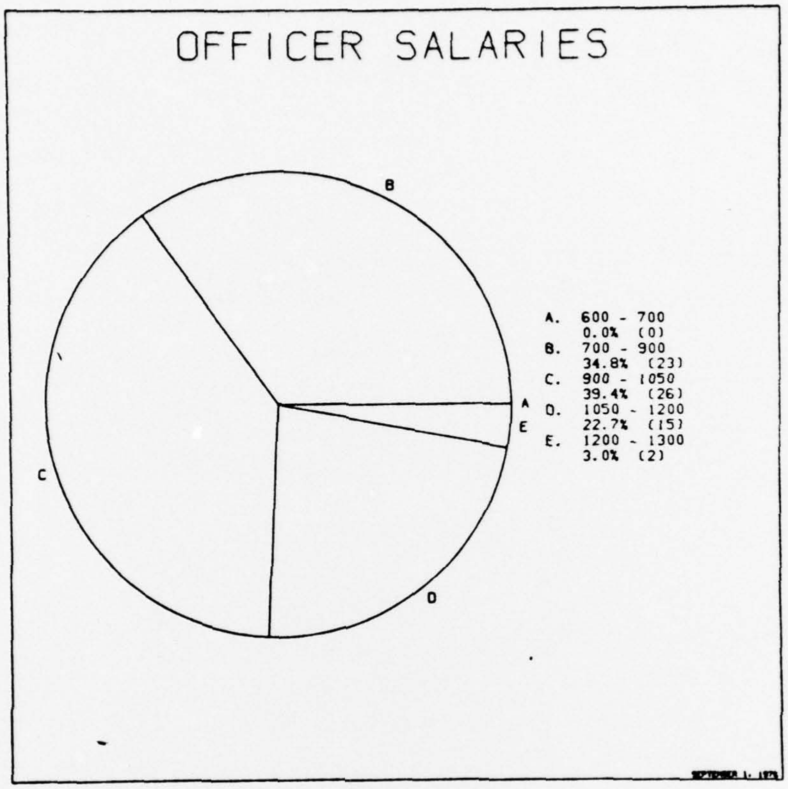
(Use of Qualifying Descriptor... This example is assumed to directly follow Example 5 such that the same STATES and annotation parameters remain in effect even though the type of plot has been changed.)

```
      .
      .
      .
      .
      .
HISTØ:  SALARY, WITH, RANK,
        2LT, 1LT, CPT, MAJ, LTC *

CØUNT:  5 *

MAIN TITLE:  OFFICER SALARIES *

YTITLE:  MONTHLY SALARY *

PLØT:
```



EXAMPLE 7

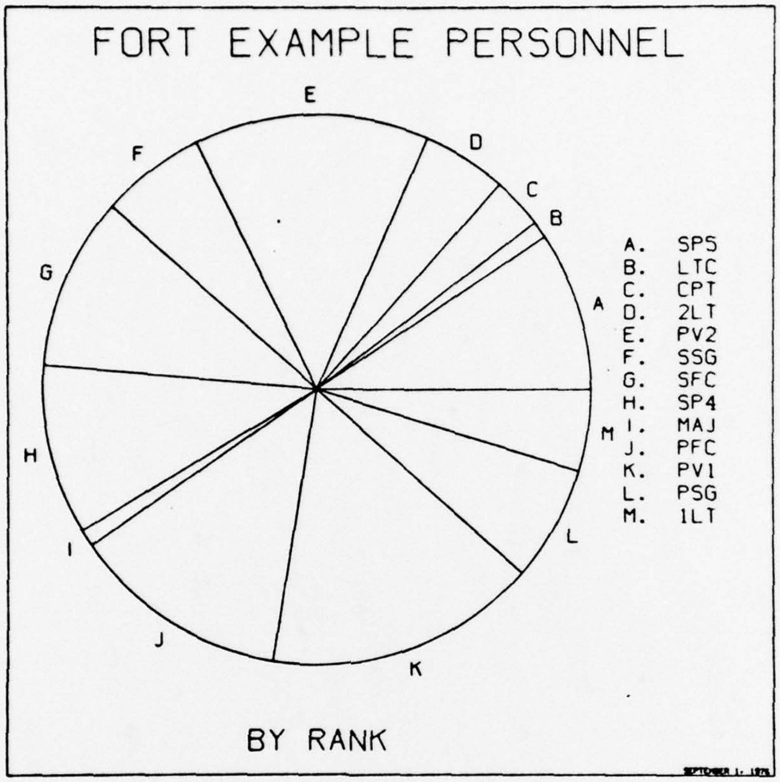
```
PIE CHART:  SALARY, WITH, RANK,
            2LT, 1LT, CPT, MAJ, LTC *
STATES:  600, 700,
         700, 900,
         900, 1050,
         1050, 1200,
         1200, 1300 *
DATE:  SEPTEMBER 1, 1976 *
MAIN TITLE:  OFFICER SALARIES *
PLØT:
```



<u>EXAMPLE 8</u>

(These commands are assumed to directly follow <u>Example 7</u>.)

```
    .
    .
    .
    .
PERCT:
QUANT:
PLOT:
```



EXAMPLE 9

```
PIE CHART:  RANK *
STATES:  .ALL. *
MAIN TITLE:  FORT EXAMPLE PERSONNEL *
XTITLE:  BY RANK *
DATE:  SEPTEMBER 1, 1976 *
PLØT:
```

APPENDIX C - THE DEVICE DEPENDENT GRAPHICS DRIVERS

The CSC plot package is output device independent. The Graphics Generation Module writes to a "logical device" which has the following characteristics:

1) The "logical" screen coordinates must be within the limits

$$-1 \leq x \leq 1$$

$$-1 \leq y \leq 1$$

2) The operations allowed on the device are

   a) MOVE to the specified screen coordinates

   b) DRAW to the specified screen coordinates

   c) Make a POINT (dot) at the specified screen coordinates

   d) FINalize the plot.

The four "logical" operations are invoked by calling the four subroutines. MOVE, DRAW, POINT, and FIN. These four subroutines constitute the Device Dependent Graphics Driver and must be loaded with the system at initialization. Loading a different set of subroutines allows the user to use a different output device.

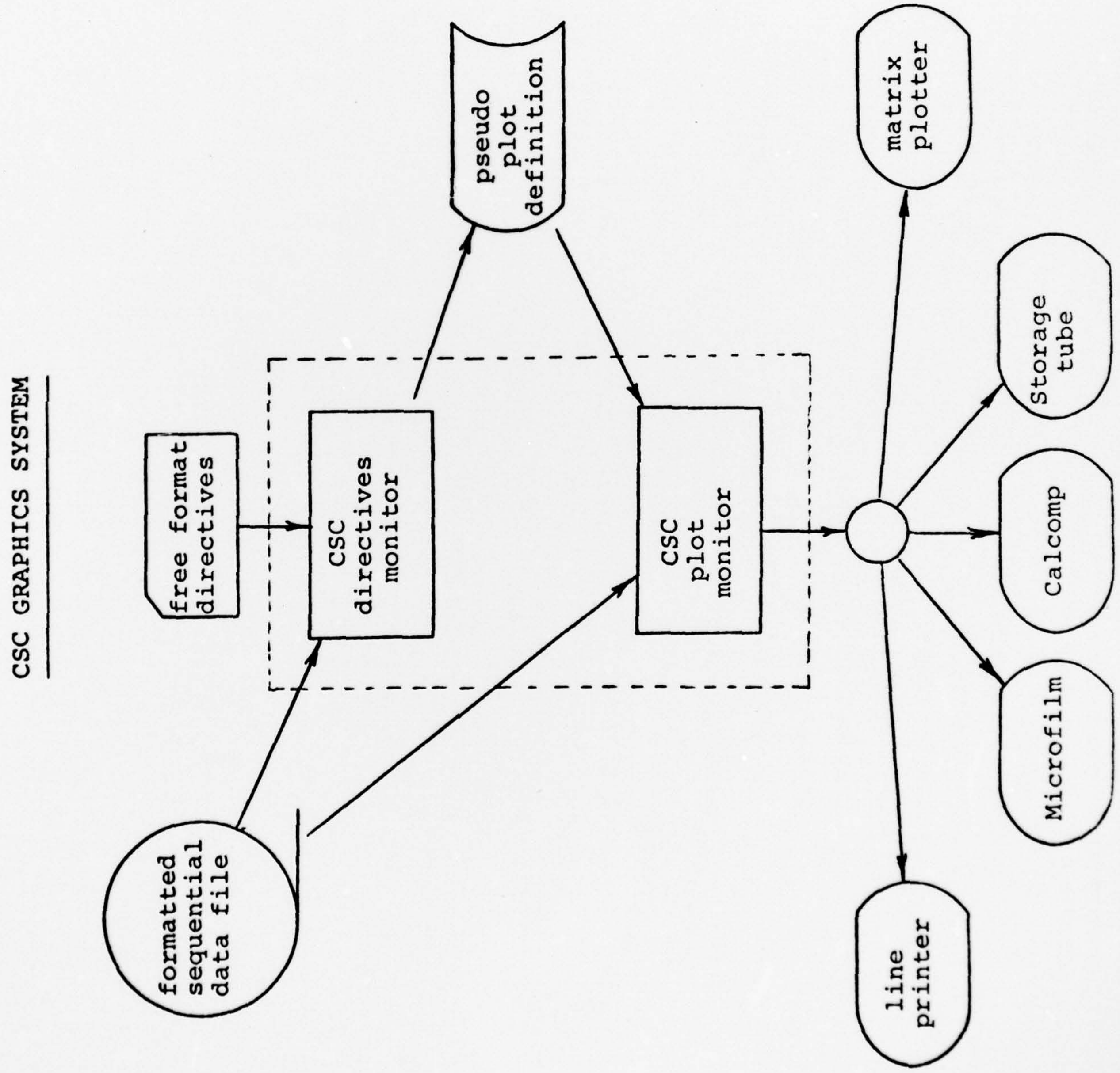
The subroutines must contain a conversion routine which changes "logical" screen coordinates to "actual" screen coordinates. They must also contain the device dependent code enabling the output device to MOVE, DRAW, draw a visible POINT, and to FINalize the plot.

A sample Driver is listed on the following pages.

Example Device Dependent Graphics Driver for the Calcomp Plotter

```
      SUBROUTINE MOVE ( X,Y )
C
C...  ROUTINE TO MOVE THE PEN TO ( X,Y )
C
C...  COMMON FLAG TO INDICATE IF THE DEVICE
C...  INITIALIZATION IS TO BE ACCOMPLISHED
C
      COMMON/INITT/INIT
      LOGICAL INIT
C
C...  COMMON AREA THAT SPECIFIES THE
C...  MINIMUM TEXT HEIGHT THAT CAN BE VISUALIZED
C...  ON THE DEVICE --- AND THE MAXIMUM HEIGHT
C...  ( HEIGHT OF THE MAIN TITLE )
C
      COMMON/MTSIZE/SIZMIN,SIZMAX
C
C...  DATA INITIALIZATION
C
      DATA SIZMIN/.Ø13/,
     .     SIZMAX/.Ø7/,
     .     INIT/.TRUE./
C
C     ..........
C
      IF ( INIT ) CALL DINIT
C
C...  CONVERT TO SCREEN COORDINATES
C
      CALL CNVERT ( X,Y,X1,Y1 )
C
C...  CALL DEVICE DRIVER TO MOVE TO ( X1,Y1 )
C...  WITH THE PEN UP.  THIS IS DEVICE DEPENDENT CODE.
C
      CALL CPLOT ( X1,Y1,3 )
C
      RETURN
C
      END
      SUBROUTINE DRAW ( X,Y )
C
C...  ROUTINE TO DRAW WITH THE PEN TO ( X,Y )
C
C...  INITIALIZATION FLAG
C
      COMMON/INITT/INIT
      LOGICAL INIT
C
C     ..........
C
      IF ( INIT ) CALL DINIT
C
C...  CHANGE TO SCREEN COORDINATES
C
      CALL CNVERT ( X,Y,X1,Y1 )
C
C...  CALL DEVICE DRIVER TO MOVE TO ( X1,Y1 )
C...  WITH THE PEN DOWN.  THIS IS DEVICE
C...  DEPENDENT CODE.
C
      CALL CPLOT ( X1,Y1,2 )
C
      RETURN
C
      END
      SUBROUTINE POINT ( X,Y )
C
C...  ROUTINE TO MAKE A DOT AT ( X,Y )
C
C...  A SMALL PLUS IS USED TO SHOW
C...  A DOT ON THE CALCOMP PLOTTER
C
C...  INITIALIZATION FLAG
C
      COMMON/INITT/INIT
      LOGICAL INIT
C
C     ..........
C
      IF ( INIT ) CALL DINIT
C
C...  CHANGE TO SCREEN COORDINATES
C
      CALL CNVERT ( X,Y,X1,Y1 )
C
C...  DEVICE DEPENDENT CODE TO
C...  GENERATE A SMALL PLUS
C
      CALL CPLOT ( X1,Y1-.Ø1,3 )
      CALL CPLOT ( X1,Y1+.Ø1,2 )
      CALL CPLOT ( X1-.Ø1,Y1,3 )
      CALL CPLOT ( X1+.Ø1,Y1,2 )
C
      RETURN
C
      END
      SUBROUTINE DINIT
C
C...  ROUTINE TO INITIALIZE THE CALCOMP
C...  PLOTTER AT THE UNIVERSITY OF COLORADO
C
C...  INITIALIZATION FLAG
C
      COMMON/INITT/INIT
      LOGICAL INIT
C
C...  DEVICE DEPENDENT CODE TO
C...  BEGIN THE PLOT
C
      CALL PSTART ( 2Ø.,4LPLOT )
C
      INIT = .FALSE.
C
      RETURN
C
      END
      SUBROUTINE CNVERT ( X,Y,X1,Y1 )
C
C...  DEVICE DEPENDENT CODE TO  CONVERT (X,Y)
C...  FROM
C...       X BETWEEN -1 AND 1
C...       Y BETWEEN -1 AND 1
C...  TO
C...       X BETWEEN Ø AND 13
C...       Y BETWEEN 3 AND 16
C
C      ..........
C
      X1 = ( X + 1. ) / 2.
      Y1 = ( Y + 1. ) / 2.
      X1 = X1 * 13.
      Y1 = Y1 * 13. + 3.
C
      RETURN
C
      END
      SUBROUTINE FIN
C
C...  FINALIZE PLOT ROUTINE
C
C...  INITIALIZATION FLAG
C
      COMMON/INITT/INIT
      LOGICAL INIT
C
C...  DEVICE DEPENDENT CODE TO
C...  END THE PLOT
C
      CALL PEND ( 21. )
      INIT = .TRUE.
C
      RETURN
C
      END
```

APPENDIX D - OVERALL SYSTEM SCHEMATIC


CSC GRAPHICS SYSTEM
formatted sequential data file
free format directives
CSC directives monitor
pseudo plot definition
CSC plot monitor
line printer
Microfilm
Calcomp
Storage tube
matrix plotter